On Re-use of Randomness in Broadcast Encryption

Paolo Gasti
Department of Information and Computer Science
University of California
Irvine (CA), USA
pgasti@uci.edu

Alessio Merlo
Dipartimento di Informatica, Sistemistica e Telematica
University of Genova
16145, Genova (Italy)
alessio.merlo@dist.unige.it

***Abstract*—Broadcast encryption provides an efficient way to encrypt a message for a large number of receivers. This paper investigates whether it is possible to further improve efficiency of an existing state-of-the-art broadcast encryption scheme by reusing some of the random choices among different encryptions, without compromising the security of the original scheme. We introduce two schemes: the first allows a transmitter to efficiently encrypt several messages to a set of users; the second scheme extends the first by allowing the transmitter to efficiently send independent messages to different groups at once. We illustrate two scenarios where our schemes provide significant advantages compared to existing solutions.**

I. Introduction

Broadcast encryption allows a centralized transmitter to efficiently encrypt messages to a (possibly large) set of users, only a privileged subset of which is allowed to decrypt them. Both the set of all users and the privileged subset are expected to vary over time. Such schemes have found applications in several different areas, such as multimedia content distribution (movies books and music), multicasting over the Internet, satellite communications and cable TV. While the obvious solution is to simply give every user its own key and transmit an individually encrypted message to every member of the privileged set, this naïve approach introduces an overhead which is linear in the size of the set of privileged users. As an alternative, a secret key can be distributed to every possible subset of users. In this case the size of the encrypted messages is clearly optimal, but each user must store a huge number of keys, resulting impractical even for rather small groups.

The primary motivation for broadcast encryption is to provide a suitable *middle ground* between these two approaches [1]. Current state-of-the-art broadcast encryption schemes offer a good trade-off between ciphertext size and the size of encryption and decryption keys, providing efficient mechanisms to dynamically add or revoke users from the privileged set. Depending on the size and characteristics of the sets of privileged and non-privileged users, different a broadcast encryption scheme can provide very different levels of efficiency. However, they all focus on sending a single message to a set of users. If the transmitter is willing to send several messages to distinguished sets of users, the only way to reach his goal is to broadcast each message separately.

We believe that there are more efficient ways to perform this task. In this paper, we propose the re-use of randomness to increase space and time efficiency of current broadcast encryption schemes. Our schemes are more efficient that current solutions when they are used to encrypt several messages at once for either the same or different sets of privileged users. In particular, our solutions are an attempt to use batching to improve efficiency. We believe that the circumstances where our schemes provide a clear advantage compared to the current state-of-the-art are common. The natural application for our scheme is in the classic secure communication setting, with the only difference that the sender is willing to send more than one encrypted message. We point out that this is not the only circumstance in which our schemes outperform current schemes. To show that, we introduce two realistic scenarios where our approach can provide substantial benefits in terms of efficiency. In particular we consider an encrypted file system and a grid access control system as our target functionality.

Encrypted file systems are used to enforce read access policies. There are basically two types of disk encryption: full disk encryption, such as Microsoft BitLocker [3] or Truecrypt [4], and “single file” disk encryption, like Windows Encrypted File System (EFS) [7]. While the first solution is appropriate for a single-user systems, the second is more suitable for multi-user environments since it allows different users to securely share some of their files with each other, while preserving data privacy from unauthorized users. In this paper we are interested in the “single file” disk encryption scenario. With EFS, each file F is encrypted using an encryption key k_F, which is stored in the file header. In order to provide data privacy, k_F is encrypted under the public key of all the users who have access to F. It is well known that this can be viewed as an application for broadcast encryption (see e.g. [17]) where the file system is the broadcast channel and the key k_F is broadcast through the file header to the broadcast set, that corresponds to the set of users who are allowed to access F. Similarly to EFS, in many encrypted file systems (see e.g. SiRiUS [5] and Plutus [6]) the header grows linearly with the size of the broadcast set. For this reason encrypted file systems work well when the number of users in the system is fairly small, but have a very low efficiency or become completely unusable when the number of users grows. As an example, for efficiency reasons the number of users who can be given access to an encrypted file using Microsoft EFS is, on average, 800 [7]. In a large organization with, say, 250,000 users this limit may not be tolerable. The use of broadcast encryption can overcome all these drawbacks, by reducing the size of the header significantly. Our approach further reduces the size of such header by exploiting the secure re-use of randomness across the encryption of different files, and reduces the time required for encryption by pre-computing part of the ciphertext once for all encrypted files.

Another suitable and unexploited use of broadcast encryption is related to the access to Grid resources. Currently, some proposals have been made to turn Service-Oriented Grid Computing in a complete cooperative platform, where single Grid users can share their permissions for granting the access to virtual resources with other users. For instance, the Cooperative Access Control (CAC) model [8] has been proposed to this aim. The idea is that each user in a virtual organization can build a *dynamic group* on-the-fly, as a repository of sharable permissions. Other users can join and share permissions in the group, in order to gain access to grid resources using permissions of other users. CAC model has been proposed in order to overcome the limitation of the basic grid access control in Grid virtual organizations which is statically managed in a centralized way by physical administrators, resulting unsuitable for supporting on-the-fly and transient cooperations among Grid users.

In the CAC model, security and performance issues arise whether all users in the virtual organization are not fully trusted and the dimension of dynamic groups is huge. In detail, security issues are related with the potential misuse of the shared permissions. At present, there exist no mechanisms in the native model that prevent the use of a permission acquired in a group outside the group itself (e.g. once a user leaves the group). On the other hand, performance issues regard the management of the group itself. In current implementations of CAC model on Grid (e.g. [9]), the group itself delivers the permission to a requesting user as soon as it asks for it. Clearly, in groups with a large number of users and permissions, the group can become a bottleneck, limiting scalability. We believe that both security and performance can be obtained through proper use of broadcast encryption, and in particular exploiting the properties of the schemes that we introduce in the following.

Our contribution We investigate whether it is possible to reduce the ciphertext size and the computational cost of current broadcast encryption schemes when several messages are broadcast to the same or different groups of privileged users at once. We introduce the appropriate adversary model for the two cases and design two corresponding broadcast encryption schemes, both based on the scheme of Delerablée et al. [10], and we provide formal proofs of security. We point out that the functionality obtained with our multi-group broadcast encryption scheme cannot be achieved using a regular broadcast encryption scheme and a key derivation function to compute different keys for different groups. Finally, we show how to efficiently apply our schemes to the scenarios previously described in this section.

II. Related Work

Broadcast encryption was first introduced by Fiat and Naor [13]. In their seminal paper they propose the first formal study of the topic and the first schemes suitable for generic broadcast settings. Their paper presents a solution that is secure against a collusion of t users, and the ciphertext size of their scheme is $\mathcal{O}(t \log^2 t \log n)$.

Naor et al. [14] presented a broadcast encryption scheme which is fully collusion secure. Their scheme is efficient when the transmitter broadcasts a message to all but a small set of users. The ciphertext size is $\mathcal{O}(r)$ when broadcasting to $n-r$ users. The size of the private keys is $\mathcal{O}(\log n)$. However this scheme does not support public-key encryption. Dodis and Fazio [15] later refined the scheme of Naor et al. into a broadcast encryption scheme with fixed size encryption key.

Boneh et al. [17] proposed a very efficient public-key broadcast encryption system where both ciphertexts and private keys are of constant size while the public key has size $\mathcal{O}(n)$. In order to decrypt ciphertexts, users need to store this public key in addition to their private keys. The same authors also introduce a second system with a different trade-off between key size and ciphertext sizes.

Delerablée et al. [10] introduced a fully collusion secure broadcast encryption scheme which is proven secure in the standard model. Their scheme allows users to join or leave without the need to alter the decryption keys. The scheme is very efficient when the size of the subset of revoked users is small: the ciphertext size is $\mathcal{O}(r)$ while the size of the decryption key is optimal, i.e. $\mathcal{O}(1)$. They also introduce another construction where the size of the ciphertext is constant while the size of the public key is $\mathcal{O}(n)$.

In general, broadcast encryption can be seen as a particular customization of a multi-recipient encryption scheme (MRES). However, the two approaches are different since they have a different focus. In general, MRES is a way to use batching in order to improve efficiency of a standard encryption scheme when used to encrypt several messages at once, while broadcast encryption allows a single broadcaster to deliver one message to a specific subset of the receivers. In broadcast encryption all the users in the privileged sets receive the same message, while MRES allows the sender to select a different message for each receiver. Moreover, key generation in a broadcast encryption scheme is performed by the transmitter, while in MRES each user generates and publishes its keys in a standard way. Note that it is possible to transform a MRES scheme into a broadcast encryption scheme, as detailed in [12].

Bellare et al. [11] introduced randomness re-use to significantly reduce time required for encryption of a message in some widely used cryptosystems. In their paper they allow the sender to maintain some state information, which is re-used across different encryptions. While the space-efficiency was not improved, the number of exponentiations needed for encryption was reduced by a factor of two or more, depending on the scheme used.

Randomness re-use has been also studied in secure multi-recipient encryption schemes. It was first introduced by Bellare et al. in [12] as a way to construct a multi-recipient scheme from a standard encryption scheme. Their work is based on the observation that randomness re-use is allowed by the use of different recipient public keys. The idea was to create a separate ciphertext for each receiver, but with the re-use of random coins in order to save bandwidth. The size of the ciphertext for the schemes they introduce is $\mathcal{O}(n)$, i.e. re-using randomness allows them to reduce the size of the ciphertext by a constant factor.

Although previous results in MRES are particularly interesting, at present no scheme has been proposed for re-using randomness in the broadcast encryption setting.

III. Preliminaries

According to the KEM-DEM paradigm, we assume that broadcast encryption schemes are used to encrypt random session keys for a symmetric encryption scheme. This standard assumption allows us to relax the security requirements for our schemes [2].

We denote by $\mathcal{U}$ the set of users listening on the broadcast channel. Let $\mathcal{R} \in \mathcal{U}$ the set of revoked users, namely the users that in a given moment are not allowed to receive the information sent by the transmitter on the broadcast channel.

A. Broadcast Encryption

A public key broadcast encryption scheme is defined by the following algorithms:

Setup(λ): a probabilistic algorithm that takes in input the security parameter λ and outputs the private key mk and the public parameter ek for the broadcast encryption scheme.

Join(mk, i): a probabilistic algorithm that takes in input the secret key mk and a user's index i and outputs a decryption key dk_i.

Encrypt$(ek, \mathcal{R})$: a probabilistic algorithm that takes in input the key ek and the set of revoked users $\mathcal{R}$ and outputs a random element K and its encryption C.

Decrypt$(dk_i, \mathcal{R}, C)$: a deterministic algorithm that takes in input a ciphertext C, a subset $\mathcal{R} \subseteq \mathcal{U}$ and a decryption key dk_i. If $i \in \mathcal{U} \backslash \mathcal{R}$, the algorithm outputs the plaintext corresponding to C.

B. Bilinear Pairings

The broadcast encryption schemes in this paper utilize bilinear maps. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of large prime order q. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficiently computable function, which must satisfy the following properties:

1) *bilinear*: $e(g^a, h^b) = e(g, h)^{ab}$ for any $a, b \in \mathbb{Z}_q$ and $g, h \in \mathbb{G}$.
2) *non-degenerate*: if g generates $\mathbb{G}$, $e(g, g)$ generates $\mathbb{G}_T$.
3) *efficient*: there is an efficient algorithm to compute $e(g, h)$ for any $g, h \in \mathbb{G}$.

C. The scheme of Delerablée et al.

Construction 1.

Setup(λ): constructs a bilinear map system $\langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$ such that $|p| = \lambda$ and pick a random generator $g \in \mathbb{G}$. Also, a secret value $\gamma \leftarrow \mathbb{Z}_p^*$. The secret key mk is set as $mk = (g, \gamma, \mathbb{S})$ where $\mathbb{S}$ is a short description of $\langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$; the encryption key is $ek = (\mathbb{S}, g, W, V)$ where $W = [\lambda]g$ and $V = e(g, g)$.

Join(mk, i): given mk and a user counter i, pick a random $x_i \leftarrow \mathbb{Z}_p^*$; set

$$A_i = \left[\frac{x_i}{\gamma + x_i}\right] g,\ B_i = \left[\frac{1}{\gamma + x_i}\right] g,\ V_i = V^{\frac{1}{\gamma + x_i}}$$

and set $dk_i = (\mathbb{S}, x_i, A_i, B_i)$ and $lab_i = (x_i, V_i, B_i)$. Note that lab_i is required to efficiently calculate the term $\frac{1}{\prod_{i \in \mathcal{R}}(\gamma + x_i)}$

Encrypt$(ek, \mathcal{R})$: The broadcaster picks $k \leftarrow \mathbb{Z}_p^*$ and computes

$$P_r = \left[\frac{1}{\prod_{i \in \mathcal{R}}(\gamma + x_i)}\right] g$$

then sets $C_1 = [k]W$ and $C_2 = [k]P_r = \left[\frac{k}{\prod_{i \in \mathcal{R}}(\gamma + x_i)}\right] g$ and outputs $C = (C_1, C_2, (x_i, P_i))$ with $i \in \mathcal{R}$

The session key is $K = K'^k$ where $K' = V^{\frac{1}{\prod_{i \in \mathcal{R}}(\gamma + x_i)}}$

Decrypt$(dk_i, \mathcal{R}, C)$: The i-th user computes

$$K = e(C_1, B_{i,\mathcal{R}}) \cdot e(A_i, C_2)$$

where

$$B_{i,\mathcal{R}} = \left[\frac{1}{\prod_{j \in \mathcal{R}}(\gamma + x_j)}\right] B_i = \left[\frac{1}{(\gamma + x_i)\prod_{j \in \mathcal{R}}(\gamma + x_j)}\right] g$$

D. Complexity Assumptions

We base the security of our scheme on a generalization of the Diffie-Hellman exponent assumption also used by Delerablée et al. in [10]. This assumption was adapted from a work of Boneh et al. [16], who first introduced it for symmetric and asymmetric bilinear map group systems. Delerablée et al. extended it to the dissociate case. In this section we provide a brief overview of the GDHE assumption.

Let $\mathbb{S} = \langle p, \mathbb{G}, \mathbb{G}_T, e\rangle$ be a bilinear map group system. Let g be a generator of $\mathbb{G}$ and $G = e(g, g) \in \mathbb{G}_T$. Let s, m be positive integers and P, $Q \in \mathbb{F}_p[X_1, \ldots, X_m]^s$ be two s-tuples of m-variate polynomials over $\mathbb{F}_p$. Set $P = (p_1, p_2, \ldots, p_s)$ and $Q = (q_1, q_2, \ldots, q_s)$ where $p_1 = 1$ and $q_1 = 1$. For any function $h : \mathbb{F}_p \to \Omega$ and vector $(x_1, \ldots, x_m) \in \mathbb{F}_m^p$, we denote $\mathrm{L}(h(p_1(x_1, \ldots, x_m)), \ldots, h(p_s(x_1, \ldots, x_m))) \in \Omega^s$ with $h(P(x_1, \ldots, x_m)))$. We denote the s-tuple Q with a similar notation. Let $F \in \mathbb{F}_p[X_1, \ldots, X_m]$. It is said that F depends on (P, Q), which we denote by $F \in \langle P, Q\rangle$, when there exists a linear decomposition

$$F = \sum_{1 \leq i,j \leq s} (a_{i,j} p_i p_j) + \sum_{1 \leq i \leq s} (b_i q_i)$$

with coefficients $a_{i,j}$, $b_i \in \mathbb{Z}_p$. Let P,Q be as above and $F \in \mathbb{F}_p[X_1, \ldots, X_m]$.

Definition 1 ((P, Q, F)-GDHE). *Given the vector*

$$H(x_1, \ldots, x_m) = \big([P(x_1, \ldots, x_m)]g, G^Q(x_1, \ldots, x_m\big)$$

$$\in \mathbb{G}^s \times \mathbb{G}_T^s$$

compute $G^{F(x_1,\ldots,x_m)}$.

Definition 2 ((P, Q, F)-GDDHE). *Given* $H(x_1, \ldots, x_m) \in \mathbb{G}^s \times \mathbb{G}_T^s$ *as above and* $T \in \mathbb{G}_T$*, decide whether* $T = g^{F(x_1,\ldots,x_m)}$.

As a strong evidence of the hardness of the GDHE assumption, Boneh et al. provide in [16] a lower bound on the advantage of an adversary in solving the BDHE problem in the generic group model. In particular, they prove that (P, Q, F)-GDHE and (P, Q, F)-GDDHE have generic security when $F \notin \langle P, Q \rangle$. We provide reductions from our solutions to the scheme of Delerablée et al. which rely on the intractability of GDHE and GDDHE for some well-defined P, Q, F.

E. Security Definitions

We define the security experiment $\mathsf{IND\text{-}SG}$ for a single-group multi-message broadcast encryption scheme as follows:

Experiment $\mathsf{IND\text{-}SG}_{\mathcal{A},\mathcal{BE}}$

1) Run $(mk, ek, \mathbb{S}) \leftarrow Setup(1^\kappa)$. $(ek, \mathbb{S})$ is sent to $\mathcal{A}$.
2) $\mathcal{A}$ outputs the set of revoked users $\mathcal{R}$; The challenger responds with the decryption keys of the users in $\mathcal{R}$.
3) A random bit b is drawn, together with two random element k_0, k_1 from the session-key space.
4) A challenge $C = (c_b, \mathcal{R}, k_0, k_1)$ is constructed, where c_b encrypts k_b, for the group of users $\mathcal{U} \setminus \mathcal{R}$. The challenge is sent to $\mathcal{A}$.
5) $\mathcal{A}$ outputs b' and the experiment outputs 1 iff $b = b'$ (i.e. iff $\mathcal{A}$ wins the experiment).

Definition 3 (semantic security for a single-group broadcast encryption scheme). *Given a security parameter* κ*, a single-group broadcast encryption scheme* $\mathcal{BE}$ = (***Setup**, **Join** , **Encrypt** , **Decrypt***) *has indistinguishable encryptions under chosen plaintext attack if there exists a negligible function* negl *such that for any probabilistic polynomial time* $\mathcal{A}$, $\Pr[\mathsf{IND\text{-}SG}_{\mathcal{A},\mathcal{BE}}(\kappa) = 1] \leq 1/2 + \mathsf{negl}(\kappa)$.

We define the security experiment $\mathsf{IND\text{-}MG}$ for a multi-group multi-message broadcast encryption scheme similarly, as follows:

Experiment $\mathsf{IND\text{-}MG}_{\mathcal{A},\mathcal{BE}}(\kappa)$

1) Run $(mk, ek, \mathbb{S}) \leftarrow Setup(1^\kappa)$. $(pk, \mathbb{S})$ is sent to $\mathcal{A}$.
2) Adversary $\mathcal{A}$ chooses two sets of revoked users $\mathcal{R}_1, \mathcal{R}_2$ and decides which one he wants to be challenged on. Let $\mathcal{A}$'s choice be $\alpha \in \{1, 2\}$.
3) The challenger responds with the decryption keys of the users in $\mathcal{R}_{3-\alpha}$
4) A random bit b is drawn, together with two random elements $k^\alpha_{1-b}, k^{3-\alpha}$ from the session-key space.
5) A challenge $C = (c_1, \mathcal{R}_1, c_2, \mathcal{R}_2, k^\alpha_0, k^\alpha_1)$ is constructed, where c_α encrypts k^α_b and $c_{3-\alpha}$ encrypts $k^{3-\alpha}$ for the respective group of users $\mathcal{U}\backslash\mathcal{R}_1$ and $\mathcal{U}\backslash\mathcal{R}_2$. The challenge is sent to $\mathcal{A}$.
6) $\mathcal{A}$ outputs bit b', and the experiment outputs 1 iff $b = b'$ (i.e. iff $\mathcal{A}$ wins the experiment).

Definition 4 (semantic security for a multi-group broadcast encryption scheme). *Given a security parameter* κ*, a multi-group broadcast encryption scheme* $\mathcal{BE}$ = (***Setup**, **Join** , **Encrypt** , **Decrypt***) *has indistinguishable encryptions under chosen plaintext attack if there exists a negligible function* negl *such that for any probabilistic polynomial time* $\mathcal{A}$, $\Pr[\mathsf{IND\text{-}MG}_{\mathcal{A},\mathcal{BE}}(\kappa) = 1] \leq 1/2 + \mathsf{negl}(\kappa)$.

IV. Our Schemes

As a warm-up, we introduce our *basic construction*, based on the scheme of Delerablée et al. Our *basic construction* features a $\mathcal{O}(r)$-size ciphertext, where r is the number of revoked users. As with the scheme of Delerablée et al., this scheme is particularly useful when the number of revoked user is small, i.e. $r < \sqrt{n}$.

Setup(λ): constructs a bilinear map system $\langle p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot)\rangle$ such that $|p| = \lambda$ and pick a random generator $g \in \mathbb{G}$. Also, select a secret value $\gamma \leftarrow \mathbb{Z}_p^*$. The secret key mk is set as $mk = (g, \gamma, \mathbb{S})$ where $\mathbb{S}$ is a short description of $\langle p, \mathbb{G}, \mathbb{G}_T, e(\cdot,\cdot)\rangle$; the encryption key is $ek = (\mathbb{S}, g, W, V)$ where $W = [\lambda]g$ and $V = e(g, g)$.

Join(mk, i): given mk and a user counter i, pick a random $x_i \leftarrow \mathbb{Z}_p^*$; set

$$A_i = \left[\frac{x_i}{\gamma + x_i}\right] g,\ B_i = \left[\frac{1}{\gamma + x_i}\right] g,\ V_i = V^{\frac{1}{\gamma + x_i}}$$

and set $dk_i = (\mathbb{S}, x_i, A_i, B_i)$.

Encrypt$(ek, \mathcal{R}, \ell)$: The broadcaster picks $k \leftarrow \mathbb{Z}_p^*$ and ℓ random values $u_1, \ldots, u_\ell \leftarrow (\mathbb{Z}_p^*)^\ell$ and computes

$$P_r = \left[\frac{1}{\prod_{i\in\mathcal{R}}(\gamma + x_i)}\right] g$$

then sets $C_1 = [k]W$ $C_2 = [k]P_r = \left[\frac{k}{\prod_{i\in\mathcal{R}}(\gamma+x_i)}\right] g$ with $i \in \mathcal{R}$ and sets $C_{u_1}, \ldots, C_{u_\ell}$ as $C_{u_j} = V^{\frac{(u_j)}{\prod_{i\in\mathcal{R}}(\gamma+x_i)}}$ and outputs $C = (C_1, C_2, (x_i, P_i), C_{u_1}, \ldots, C_{u_\ell})$.

The session keys $K_1, \ldots, K_\ell$ are calculated as $K_j = K'^{k+u_j}$ where $K' = V^{\frac{1}{\prod_{i\in\mathcal{R}}(\gamma+x_i)}}$

Decrypt$(dk_i, \mathcal{R}, C)$: The i-th user computes

$$K_j = e(C_1, B_{i,\mathcal{R}}) \cdot e(A_i, C_2) \cdot C_{u_j}$$

where

$$B_{i,\mathcal{R}} = \left[\frac{1}{\prod_{j\in\mathcal{R}}(\gamma + x_j)}\right] B_i = \left[\frac{1}{(\gamma + x_i)\prod_{j\in\mathcal{R}}(\gamma + x_j)}\right] g$$

We prove the semantic security of our schemes by providing a reduction to the broadcast encryption scheme in Construction 1 of Delerablée et al. [10]. Wlog we show that there is a reduction from a our scheme with ℓ set to 1 to the scheme of Delerablée et al.

Theorem 1. *Assuming that Construction 1 of Delerablée et al. is semantically secure, the scheme above is* $\mathsf{IND\text{-}SG}$*-secure.*

Proof: To establish the semantic security of our scheme we assume to be given an adversary $\mathcal{A}$ which breaks our scheme and we build a reduction algorithm $\mathcal{B}$ that breaks the scheme of Delerablée et al. by distinguishing between a random element in $\mathbb{G}_T$ and the session key K_b (with $b \in \{0, 1\}$) corresponding to hdr given (hdr, K_0, K_1).

$\mathcal{B}$ engages the semantic security game with the challenger. $\mathcal{B}$ is given a group system $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot, \cdot))$. $\mathcal{B}$ uses $\mathbb{S}$ to setup the semantic security experiment with $\mathcal{A}$. Eventually $\mathcal{A}$ chooses a set $\mathcal{R} \subseteq \{1, \ldots, n\}$ of revoked users and $\mathcal{B}$ outputs $\mathcal{R}$ as its choice. $\mathcal{B}$ receives a set of values (x_i, A_i, B_i) with $i \in \mathcal{R}$ from the challenger and forwards it to $\mathcal{A}$. Then $\mathcal{B}$ receives the challenge ciphertext (hdr, K_0, K_1) where hdr corresponds to $(C_1, C_2, (x_1 \in \mathcal{U}, \ldots, x_r \in \mathcal{U}))$, K_b, corresponds to the decryption of hdr by a non-revoked user and K_{1-b} corresponds to a random element in $\mathbb{G}_T$. $\mathcal{B}$ constructs the challenge ciphertext for $\mathcal{A}$ as $(\mathsf{hdr}, C_{u_1}, K_0 \cdot C_{u_1}, K_1 \cdot C_{u_1})$, then sends it to $\mathcal{A}$. Eventually $\mathcal{A}$ returns its guess for b and $\mathcal{B}$ simply forwards it to the challenger.

Let us denote our scheme as $\mathcal{BE}$ and the scheme of Delerablée et al as $\mathcal{DBE}$. It is not difficult to see that $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{DBE}}(t, n, \kappa) = \mathsf{Adv}^{\mathsf{IND\text{-}SG}}_{\mathcal{BE}}(t, n, \kappa)$. That is, $\mathcal{B}$ answers the challenge correctly if and only if $\mathcal{A}$ answers its challenge correctly. The challenge ciphertext for $\mathcal{A}$ is distributed as expected, therefore the simulation is undetectable. Since $\mathcal{B}$ cannot guess correctly non-negligibly more than half of the times, $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{BE}}(t, n, \kappa) \leq 1/2 + \mathsf{negl}(\kappa)$ for some negligible function negl.

■

Note that it is possible to further reduce the ciphertext size by also considering session key K_0 as the decryption of the ciphertext using the Delerablée et al. decryption algorithm, i.e.

$$K_0 = e(C_1, B_{i,\mathcal{R}}) \cdot e(A_i, C_2)$$

In this case the ciphertext size is shorter by one element in $\mathbb{G}_T$ compared to the basic scheme.

A. A Scheme for Fast-changing Broadcast Groups

Our *basic construction* introduced in section IV allows the transmitter to send shorter ciphertexts compared to the scheme of Delerablée et al. when broadcasting several messages to the same group of users. However, as discussed in the introduction, a finer granularity may be necessary in several circumstances. In particular, we want to provide a cryptographic primitive that allows the sender to efficiently encrypt a set of messages in such a way that some of the users are only unable to decrypt some of them. Clearly the transmitter can use our previous scheme to achieve this goal by simply issuing a new set of encrypted messages for each different group of authorized users. This would, however, reduce the efficiency advantage of our scheme compared to existing solutions.

In this section we show how to encrypt a ciphertext that carries several messages, where different subsets of messages can be decrypted by different sets of privileged receivers. We introduce our *multi-group construction*, which is a generalization of our *single-group construction*. With this scheme, the sender encrypts messages for n different groups of users. For each group of users Gr_i with $i \in \{1, \ldots, n\}$, the sender encrypts ℓ_i different messages. Algorithms ***Setup*** and ***Join*** remain the same as in our previous scheme, while ***Encrypt***, ***Decrypt*** change as follows:

Encrypt$(ek, (\ell_1, \mathcal{R}_1), ..., (\ell_n, \mathcal{R}_n))$: The broadcaster picks $k \leftarrow \mathbb{Z}_p^*$ and, for each $i \in \{1, \ldots, n\}$, ℓ_i random values $u_1, \ldots, u_\ell$ from $\mathbb{Z}_p^*$ and computes

$$P_{\mathcal{R}_1} = \left[\frac{1}{\prod_{i \in \mathcal{R}_1}(\gamma + x_i)}\right] g$$

$$\cdots$$

$$P_{\mathcal{R}_n} = \left[\frac{1}{\prod_{i \in \mathcal{R}_n}(\gamma + x_i)}\right] g$$

then sets $C_1 = [k]W$, $C_{\mathcal{R}_i} = [k]P_{\mathcal{R}_i}$ and sets $C^i_{u_1}, \ldots, C^i_{u_{n_{\ell_i}}}$ as

$$C^i_{u_j} = V^{\frac{(u_j)}{\prod_{i \in \mathcal{R}_1}(\gamma + x_i)}}, \ldots, C^i_{u_1}, \ldots, C^n_{u_{n_{\ell_n}}}$$

and outputs

$$C = (C_1, C_{\mathcal{R}_1}, (x_i, P_i), C^1_{u_1}, \ldots, C^1_{u_{\ell_1}}, \ldots, C_{\mathcal{R}_n}, (x_j, P_j), C^n_{u_1}, \ldots, C^n_{u_{n_{\ell_n}}}$$

Decrypt(dk_i, C, j, m): To decrypt the m-th message for the j-th group, the i-th user computes:

$$K_j = e(C_1, B_{i,\mathcal{R}_j}) \cdot e(A_i, C_{\mathcal{R}_j}) \cdot C^j_{u_m}$$

We point out that when $\ell = 1$ this scheme is the same as the previous construction. Note that the broadcaster can use this scheme to send a single message to slightly different subsets of users. We believe that this mode is interesting in itself and has practical applications.

We provide a security proof for our *multi-group construction* below. The proof is limited wlog to the case where $\ell = 2$ and $n = 1$.

Theorem 2. *Assuming that our* basic construction *is* $\mathsf{IND\text{-}SG}$*-secure, our* multi-group construction *is* $\mathsf{IND\text{-}MG}$*-secure.*

Proof: We prove theorem 2 by reduction. Assume there exists an efficient (i.e. PPT) adversary $\mathcal{A}$ that breaks our multi-group scheme, we show that $\mathcal{A}$ can be used to construct an efficient algorithm $\mathcal{B}$ that breaks our basic scheme. $\mathcal{B}$ engages in the semantic security game for our basic scheme with the challenger and receives a groups system $\mathbb{S} = (p, \mathbb{G}, \mathbb{G}_T, e(\cdot, \cdot))$ and the set of users $\mathcal{U} = \{1, \ldots, n\}$. $\mathcal{B}$ forwards $\mathbb{S}$ and n to the adversary, which chooses two sets $\mathcal{R}_1 \subseteq \{1, \ldots, n\}$ and $\mathcal{R}_2 \subseteq \{1, \ldots, n\}$ of revoked users, outputs them to $\mathcal{B}$ and also outputs $\alpha \in \{1, 2\}$ corresponding to its choice for the set of revoked users $\mathcal{A}$ wants to be challenged upon. $\mathcal{B}$ requests the decryption keys corresponding to the users in $\mathcal{R}_\alpha$ to the challenger and returns them to $\mathcal{A}$. The challenger sends a challenge $C = (c_b, \mathcal{R}, k_0, k_1) = (C_1, C_2, (x_i, P_i), C_{u_1}, k_0, k_1)$ to $\mathcal{B}$, which uses C to construct the challenge for $\mathcal{A}$ as follows: $\mathcal{B}$ sets $\overline{C} = (C_1, C_{\mathcal{R}_1}, (x_i, P_i), C^1_{u_1}, C_{\mathcal{R}_2}, (x_j, P_j), C^2_{u_1},\ k_0, k_1)$ where $C_{\mathcal{R}_\alpha} = C_2$ and $C_{\mathcal{R}_2}$, (x_j, P_j) and $C^2_{u_1}$ are random values chosen uniformly from the proper sets.

Eventually $\mathcal{A}$ outputs its choice for b' and $\mathcal{B}$ outputs b' to the challenger. Let us denote our scheme as $\mathcal{MBE}$ and our basic scheme as $\mathcal{BE}$. It is not difficult to see that $\mathsf{Adv}^{\mathsf{IND\text{-}SG}}_{\mathcal{BE}}(t, n, \kappa) = \mathsf{Adv}^{\mathsf{IND\text{-}MG}}_{\mathcal{MBE}}(t, n, \kappa)$. That is, $\mathcal{B}$ answers the challenge correctly if and only if $\mathcal{A}$ answers its challenge correctly. The challenge ciphertext for $\mathcal{A}$ is distributed as expected, therefore the simulation is undetectable. Since $\mathcal{B}$ cannot guess correctly non-negligibly more than half of the times, $\mathsf{Adv}^{\mathsf{ind}}_{\mathcal{MBE}}(t, n, \kappa) \leq 1/2 + \mathsf{negl}(\kappa)$ for some negligible function negl.

■

Note that, as with the basic scheme, the key K_0 for each group can be used to further reduce ciphertext size by one element for each different group.

B. An Efficient Representation for the Ciphertext

The output of the encryption algorithm of Section IV-A allows sets of (x_i, P_i) to repeat, unnecessarily increasing the size of the ciphertext. In order to maximize space and time efficiency, the transmitter must avoid repetition of the sets (x_i, P_i) by combining together different representations of the same set of revoked users. This allows the transmitter to send only one element $C_{\mathcal{R}_i}$ for each set of revoked users.

It is possible to further reduce the size of the ciphertext when the groups of revoked users differ only by a few elements. An optimal strategy when the transmitter wants to add a new set of revoked users to an existing ciphertext (as in the examples we gave in the introduction) is to compare each set of revoked users already in the ciphertext with the new one. For each set $\mathcal{R}_i$ in the ciphertext, the transmitter calculates the difference between $\mathcal{R}_i$ and the current set $\mathcal{R}_j$. One possible representation for this difference is a short index for $\mathcal{R}_i$, concatenated with a list of elements $L^{i,j}_R$ that must be removed from $\mathcal{R}_i$ and a list of elements $L^{i,j}_A$ that must be added to $\mathcal{R}_i$ in order to obtain the set $\mathcal{R}_j$. Let $S^{i,j} = |L^{i,j}_A| + |L^{i,j}_R|$. The transmitter uses the representation with the smallest $S^{i,j}$. If the smallest $S^{i,j}$ is greater than $|\mathcal{R}_j|$, then the transmitter simply sends $\mathcal{R}_j$.

However, in some circumstances the list of sets of revoked used is known at once. Using the previous algorithm choosing the sets to add in random order may not be the best strategy. A more reasonable approach is the following:

1) Order the sets of revoked users by their size
2) Select one of the sets whose size is the median of the sizes of all sets
3) Add this set to the ciphertext
4) Using the algorithm above, add the set whose length is closer to the last added set
5) Repeat step 4 until all sets have been added

We do not investigate further this point since this topic is out of the scope of the paper. We leave it as an open issue for further investigation.

V. Practical Examples

In this section, we show how our schemes can be used to construct efficient solutions to the problems proposed in Section I.

A. Encrypted File System

Confidentiality is becoming a mandatory requirement in data storage systems. The low cost and ubiquitous nature of mass storage devices has increased the risk of data theft, which can be perpetrated by insiders and outsiders. Encrypted file systems (EFS) provide an effective solution to protect the confidentiality of data at rest. Moreover, they provide a solid tool for enforcing read access policies when each file is encrypted under a separate key, known only to the individuals who have a right to access that file.

Unfortunately, when an EFS is used to provide access control on a system with a large number of users, a large overhead is introduced. File systems like Microsoft EFS [7] use an hybrid approach to encrypt files. Each file is encrypted using a symmetric key, and the symmetric key is then encrypted under the public key of all users who are allowed to read that file. This solution does not scale well, since the space required to encrypt the symmetric key is linear in the number of users allowed to access a specific file.

Our multi-group approach can be used to greatly reduce this overhead. More specifically, a broadcast encryption scheme can be used, instead of a regular public-key encryption scheme, to encrypt the symmetric keys. This idea is not new (see e.g. [17]), however the efficiency of previous solutions can be significantly improved by using the multi-group broadcast encryption scheme introduced in this paper.

An efficient multi-user encrypted filesystem Users $U_1, \ldots, U_n$ share a common filesystem. Several groups $Gr_1, \ldots, Gr_m$ of users are created, where group Gr_i is defined as the set of users who don't belong to Gr_i, i.e. $Gr_i = \mathcal{R}_i$.

The file system is initialized and a header is created as the encryption ***Encrypt***$(ek, (0, \mathcal{R}_1), \ldots, (0, \mathcal{R}_m))$. When a file F belonging to group Gr_i is added to the file system, it is encrypted using a random symmetric key k_F. The key k_F is obtained from the multi-group broadcast encryption scheme:

1) an element u_F is chosen uniformly at random from $\mathbb{Z}^*_P$
2) u_F is used to construct the element C_{u_F} as with the ***Encrypt*** algorithm of our multi-group broadcast encryption scheme
3) C_{u_F} is added to the file system header, and the key $k_f = K'^{k+u_F}$ is returned

Our multi-group scheme is more efficient, compared to existing broadcast encryption schemes: one single element of $\mathbb{G}_T$ corresponds to the encryption of a file, rather than two as in the state-of-the-art scheme of Delerablée et. al. [10] – and time. Our scheme is also time-efficient since part of the encryption of a ciphertext is precomputed when the file system is created and the header is initialized.

We assume that the total number of groups of users is significantly smaller than the number of users. With our scheme we expect the size of the elements needed to represent the groups to be negligible compared to the size of the file system. Moreover, the strategies in Section IV-B can be used to further reduce the size of the file system header.

B. CAC on Grid through Dynamic Groups

Dynamic Groups implement the CAC model and can be exploited for resource sharing within a Grid Virtual Organization (VO). In a simple way, a VO is made of a set of Grid Users $\{GU_i\}$ and a set of Virtual Resources ($\{VR_j\}$). Each GU_i disposes of a set of statically defined permissions ($\{P_{i,j}\}$) allowing her to access some resources vr_j. The access control is identity-based, thus, any $GU_{i'}$ that possesses $P_{i,j}$ and requires to access VR_j is recognized as GU_i by the resource[1]. In the CAC model, a GU provides (1) its identity though temporary certificates (e.g. Proxies) and (2) a list of VRs that are exploitable through her identity to a shared and transient repository (i.e. dynamic group). If another GU aims to access a shared resource, she can require to join the group and, once allowed, she can query the group for the related permission receiving back one identity certificate. It is trivial to understand the importance of both security and performance issues discussed in Section I in VOs that are wide and not fully trusted.

Securing Dynamic Groups through Broadcast Encryption The definition of a proper broadcast channel, together with our single-group scheme provide a solution to the previous issues.

We define the broadcast channel in a dynamic group as the set of information accessible to all VRs in the VO and to the group users. This means that a GU is allowed to see the *channel* once the group has granted her to access and until she leaves the group.

During the lifetime of a group, periods of time during which the status of the group does not change (i.e. identical sets of GUs, VRs, and permissions) are called *time slots*.

Our single group scheme can be applied to cipher permissions with a different session key at the beginning of each time slot; the set of accessible VRs in a time slot corresponds to the set of revoked users of the original scheme. More specifically, once a group is built, given a λ_{Gr}, the execution of $\mathbf{Setup}(\lambda_{Gr})$ returns ek_{Gr} and mk_{Gr}.

Thus, given $VRSet = \{VR_j\}$ for some j the set of VR in the VO, the group executes $dk_j = Join(mk_{Gr}, j)$ and securely delivers each key to the corresponding VR_j. At the beginning of the first slot (i.e. when the group is empty) an initial configuration is built through $\mathbf{Encrypt}(ek_{Gr}, N, 1)$, where N is the cardinality of $VRSet$.

At the ℓ-th time slot, given $\mathcal{R}$ as the set of VRs that do not belong to the group at the beginning of the slot, the group executes:

1) an element u_ℓ is chosen uniformly at random from $\mathbb{Z}_P^*$
2) u_ℓ is used to build $C_{u-\ell}$ through the $\mathbf{Encrypt}(ek_{Gr}, \mathcal{R}, \ell)$ algorithm.
3) $C_{u-\ell}$ is added to the initial configuration and $k_{u_\ell} = K'^{k+u_\ell}$ is used to cipher permissions.

Encrypted permissions, C, $\mathcal{R}$ and the list of shared VRs are published in the group. Each authenticated GU accesses the broadcast channel and retrieves the required permission; then she supplies it to the VR. The VR accesses the channel and obtains C and $\mathcal{R}$ for deciphering the permission through the **Decrypt** function.

The proposed algorithm, together with our single-group scheme, form a possible solution for the security and performance issues related with dynamic groups. In particular, the use of both encryption of permissions and time slots allows a user to exploit shared permissions as long as she belongs to the group. Moreover, the use of a broadcast channel accessible by authorized users and resources frees the group from the burden of sending permissions on demand, potentially improving the scalability of dynamic groups.

VI. Conclusions

We introduced the first broadcast encryption scheme that takes advantage of randomness re-use to reduce the size of ciphertext when several messages are sent at once to different groups. Our scheme allows the transmitter to encrypt ℓ session keys using a roughly half the elements in $\mathbb{G}_T$ compared to the current state-of-the art scheme of Delerablée et al. [10].

Our scheme is secure in the standard model, and we provided formal proofs by reduction to an existing and widely accepted hardness assumption.

Finally, we proposed two realistic scenarios, an encrypted file system and a resource sharing technique for Grid environments, where our schemes offer substantial advantages in comparison to the existent solutions.

Acknowledgement

Part of this work was developed in conjunction with the NSF-sponsored Named Data Network (NDN) project [18].

References

[1] J. Horwitz. A Survey of Broadcast Encryption. 2003.

[2] J. Herranz, D. Hofheinz, and E. Kiltz. KEM/DEM: Necessary and Sufficient Conditions for Secure Hybrid Encryption. IACR ePrint Archive, 2006.

[3] R. Humpries. Bitlocker Drive Encryption and Disk Sanitization. TechNet, Microsoft, 11th of October, 2006.

[4] TrueCrypt Tool, www.truecrypt.org

[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh. Sirius: Securing remote untrusted storage. Proc. Network and Distributed Systems Security (NDSS) Symposium 2003, pp. 131-145.

[1] coherently with the set of $P_{i,j}$ defined, a VR stores a list of user identities that are allowed to access it

[6] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu. Plutus: Scalable Secure File Sharing on Untrusted Storage. Proceedings of the 2nd USENIX Conference on File and Storage Technologies (San Francisco, CA, March 31 - 31, 2003). Conference On File And Storage Technologies. USENIX Association, Berkeley, CA, 29-42.

[7] R. Muller. How it works: Encrypting File System. Microsoft TechNet Magazine, May 2006.

[8] A. Merlo and A. Armando. Cooperative Access Control for the Grid. Proceedings of the 6th International Conference on Information Assurance and Security (IAS 2010), Atlanta, GA, USA, August 23-24, 2010.

[9] A. Merlo. A Cooperative Model for Resource Sharing on Grid. Journal of Information Assurance and Security (JIAS), v. 6, n. 1, 2011, pp. 106-114

[10] C. Delerablée, P. Paillier, and D. Pointcheaval. Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys. Pairing-Based Cryptography Pairing 2007 (2007), pp. 39-59.

[11] M. Bellare, T. Kohno, and V. Shoup. Stateful public-key cryptosystems: how to encrypt with one 160-bit exponentiation. Proceedings of the 13th ACM Conference on Computer and Communications Security (Alexandria, Virginia, USA, October 30 - November 03, 2006). CCS '06. ACM, New York, NY, pp. 380-389.

[12] M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon. Multi-Recipient Encryption Schemes: Efficient Constructions and their Security. IEEE Transactions on Information Theory, Volume 53, Number 11, 2007.

[13] A. Fiat, M. Naor. Broadcast encryption. Proceedings of the 13th Annual international Cryptology Conference on Advances in Cryptology (Santa Barbara, California, United States). D. R. Stinson, Ed. Springer-Verlag New York, New York, NY, 1993, pp. 480-491.

[14] D. Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 4162. Springer, Heidelberg (2001)

[15] Y. Dodis and N. Fazio. Public-key Broadcast Encryption for Stateless Receivers. ACM Workshop in Digital Rights Management—DRM 2002, pp. 61-80.

[16] Boneh, D., Boyen, X., Goh, E.-J.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440456. Springer, Heidelberg (2005), Extended version available at http://eprint.iacr.org/2005/015

[17] D. Boneh, C. Gentry, and B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys Crypto, 258-275, 2005.

[18] Named Data Networking, http://www.named-data.net/